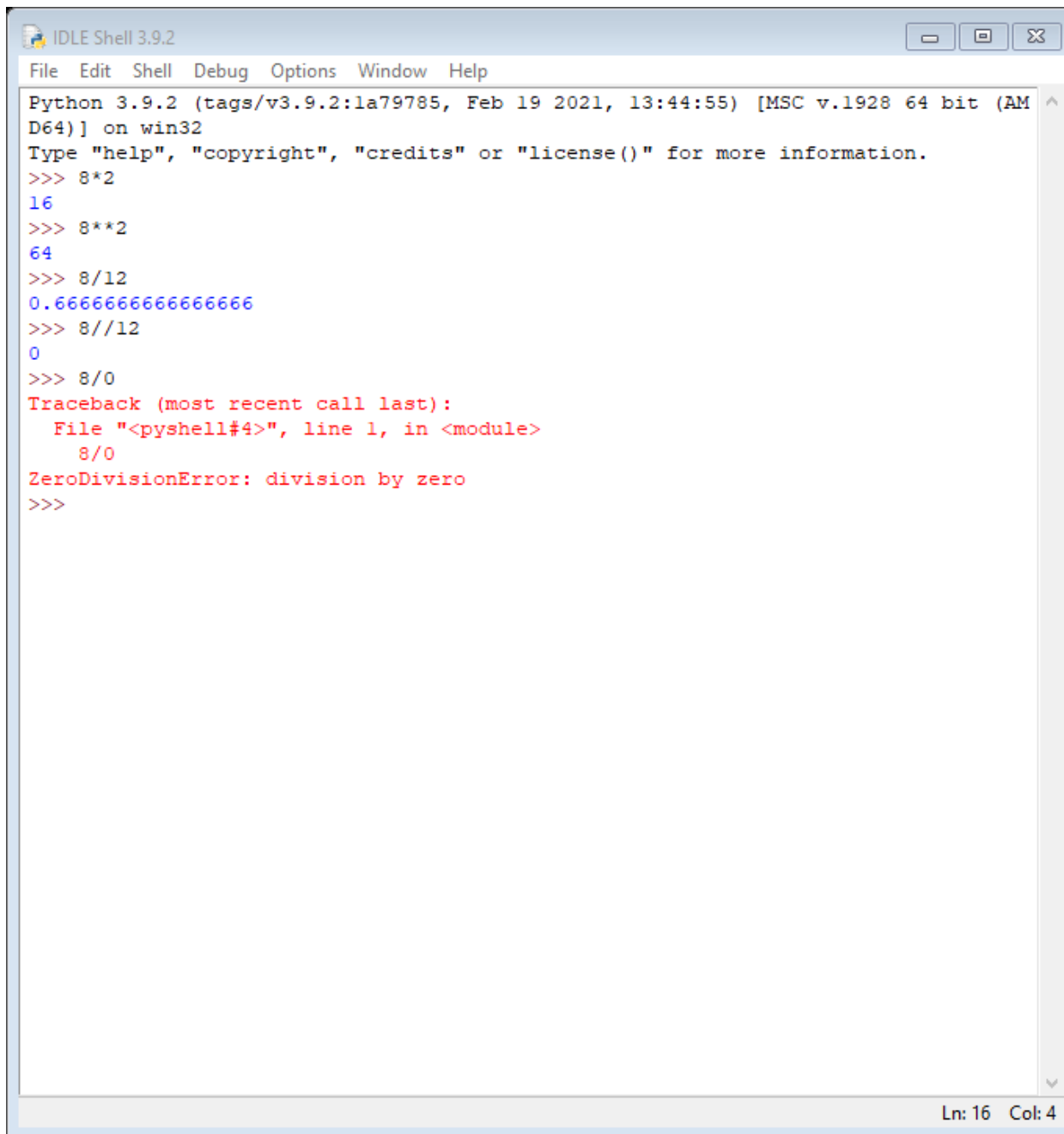


QN. 1



The screenshot shows the IDLE Shell 3.9.2 window. The title bar reads "IDLE Shell 3.9.2". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The shell area contains the following text:

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 8*2
16
>>> 8**2
64
>>> 8/12
0.6666666666666666
>>> 8//12
0
>>> 8/0
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    8/0
ZeroDivisionError: division by zero
>>>
```

At the bottom right of the window, the status bar displays "Ln: 16 Col: 4".

QN.2



The screenshot shows the Thonny IDE interface. The main editor window displays a Python function named `personal_details()` with the following code:

```
1 def personal_details():
2     name, age = "Santosh Bhandari", 19
3     address = "baneshwor, kathmandu"
4     phone = "123456789"
5     print("Name: {}\nAge: {}\nAddress: {}\nPhone number: {}".format(name, age, address, phone))
6
7 personal_details()
```

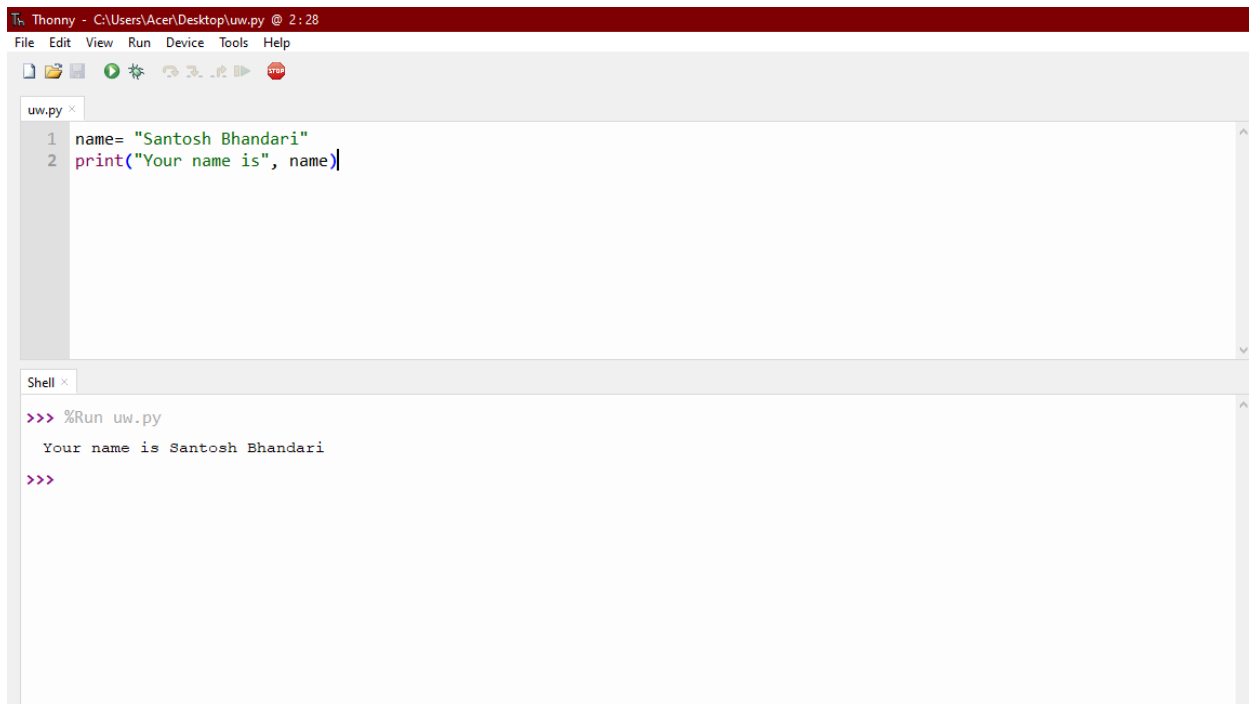
Below the editor is a Shell window showing the output of running the script:

```
>>> %Run uw.py
Name: Santosh Bhandari
Age: 19
Address: baneshwor, kathmandu
Phone number: 123456789
>>> |
```

On the right side, there is an Assistant panel with a message:

The code in `uw.py` looks good.
If it is not working as it should, then consider using some general [debugging techniques](#).
[Was it helpful or confusing?](#)

QN. 3



The screenshot shows the Thonny IDE interface. The main editor window displays a simple Python script:

```
1 name= "Santosh Bhandari"
2 print("Your name is", name)
```

Below the editor is a Shell window showing the output of running the script:

```
>>> %Run uw.py
Your name is Santosh Bhandari
>>>
```

Qn. 4



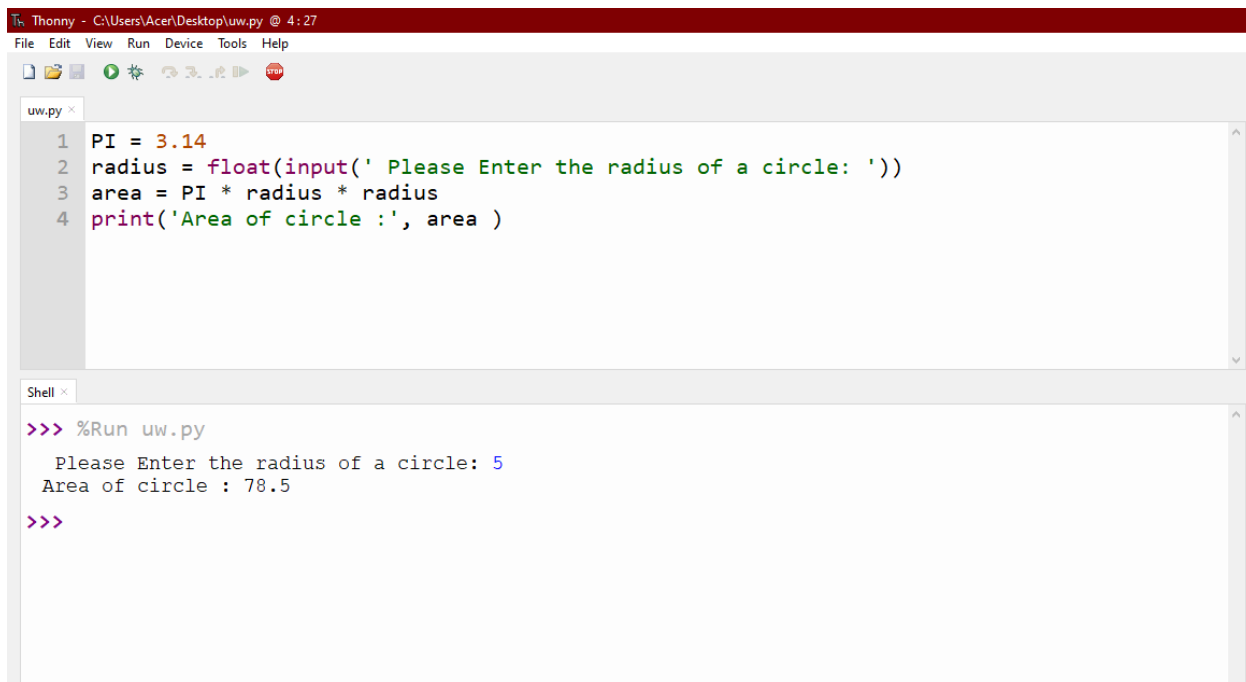
The screenshot shows the Thonny IDE interface. The top bar indicates the file path is C:\Users\Acer\Desktop\uw.py and the time is 3:11. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations, running, and debugging. The main editor window displays a Python script named uw.py with the following code:

```
1 name = input("Enter your name: ")
2
3 age = int (input("Enter your age: "))
4
5
6 print("Your name is",name,"and you are",age,"year(s) old.")
```

Below the editor is a Shell window showing the execution of the script:

```
>>> %Run uw.py
Enter your name: Santosh Bhandari
Enter your age: 19
Your name is Santosh Bhandari and you are 19 year(s) old.
>>> |
```

Qn. 5



The screenshot shows the Thonny IDE interface. The top bar indicates the file path is C:\Users\Acer\Desktop\uw.py and the time is 4:27. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations, running, and debugging. The main editor window displays a Python script named uw.py with the following code:

```
1 PI = 3.14
2 radius = float(input(' Please Enter the radius of a circle: '))
3 area = PI * radius * radius
4 print('Area of circle :', area )
```

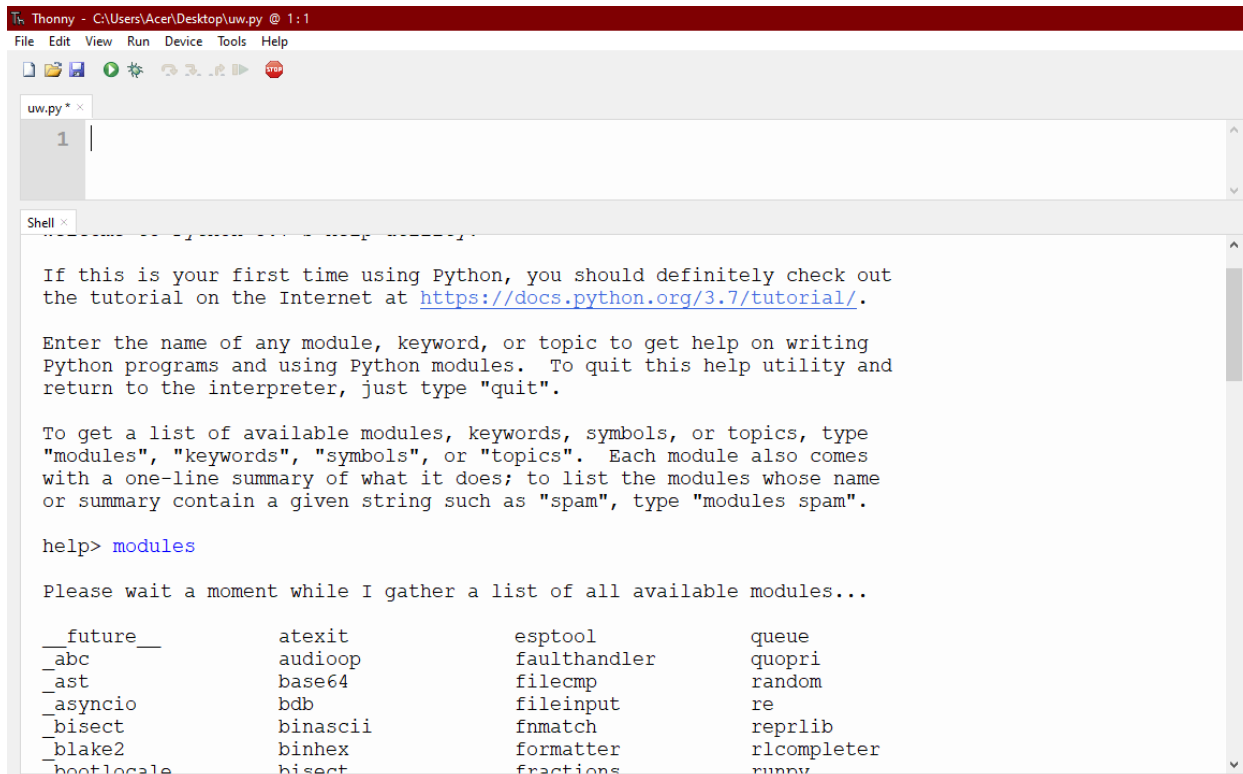
Below the editor is a Shell window showing the execution of the script:

```
>>> %Run uw.py
Please Enter the radius of a circle: 5
Area of circle : 78.5
>>>
```

Qn. 6

Explanation: This gives error at line num= num + 1 as cannot convert int object to str implicitly

Qn. 7



Thonny - C:\Users\Acer\Desktop\uw.py @ 1:1

File Edit View Run Device Tools Help

uw.py * x

```
1 |
```

Shell x

If this is your first time using Python, you should definitely check out the tutorial on the Internet at <https://docs.python.org/3.7/tutorial/>.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

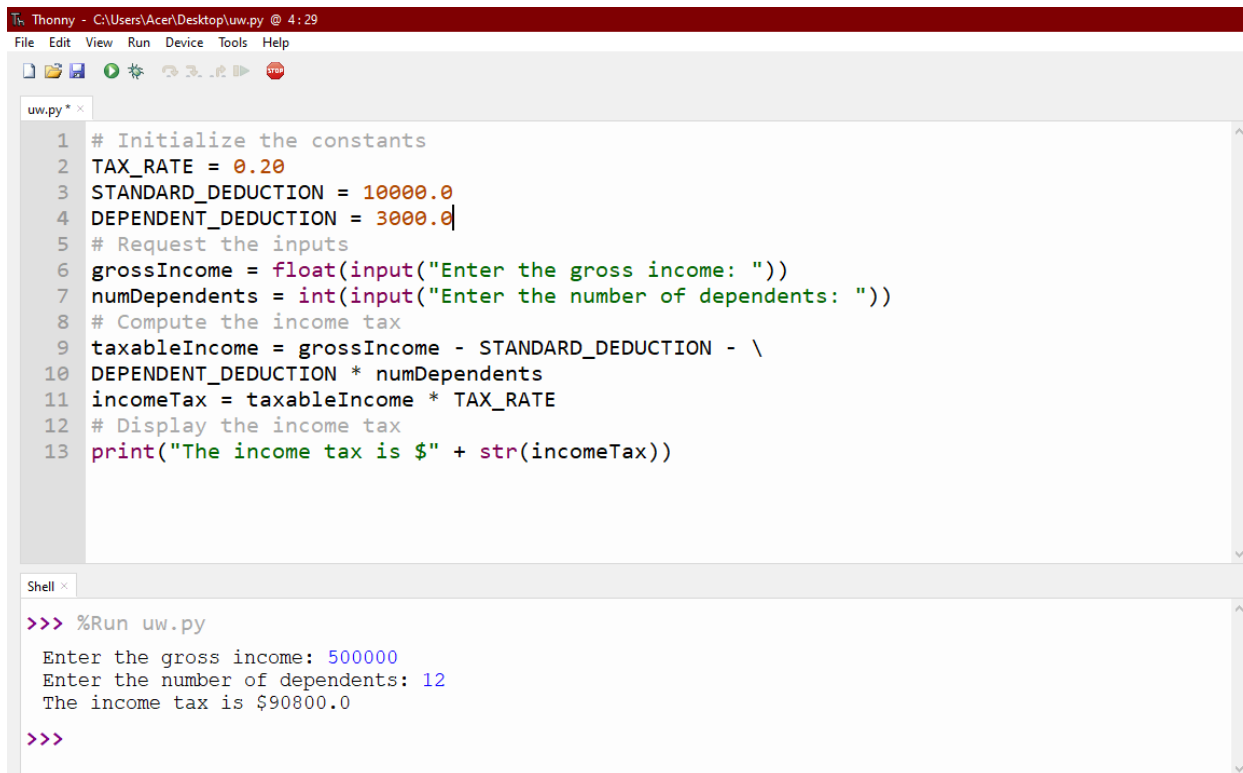
To get a list of available modules, keywords, symbols, or topics, type "modules", "keywords", "symbols", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string such as "spam", type "modules spam".

help> modules

Please wait a moment while I gather a list of all available modules...

__future__	atexit	esptool	queue
__abc__	audioop	faulthandler	quopri
__ast__	base64	filecmp	random
__asyncio__	bdb	fileinput	re
__bisect__	binascii	fnmatch	reprlib
__blake2__	binhex	formatter	rlcompleter
__bootlocale__	bisect	fractions	runpy

Qn. 8



Thonny - C:\Users\Acer\Desktop\uw.py @ 4:29

File Edit View Run Device Tools Help

uw.py * x

```
1 # Initialize the constants
2 TAX_RATE = 0.20
3 STANDARD_DEDUCTION = 10000.0
4 DEPENDENT_DEDUCTION = 3000.0
5 # Request the inputs
6 grossIncome = float(input("Enter the gross income: "))
7 numDependents = int(input("Enter the number of dependents: "))
8 # Compute the income tax
9 taxableIncome = grossIncome - STANDARD_DEDUCTION - \
10 DEPENDENT_DEDUCTION * numDependents
11 incomeTax = taxableIncome * TAX_RATE
12 # Display the income tax
13 print("The income tax is $" + str(incomeTax))
```

Shell x

```
>>> %Run uw.py
Enter the gross income: 500000
Enter the number of dependents: 12
The income tax is $90800.0
>>>
```

Qn. 9



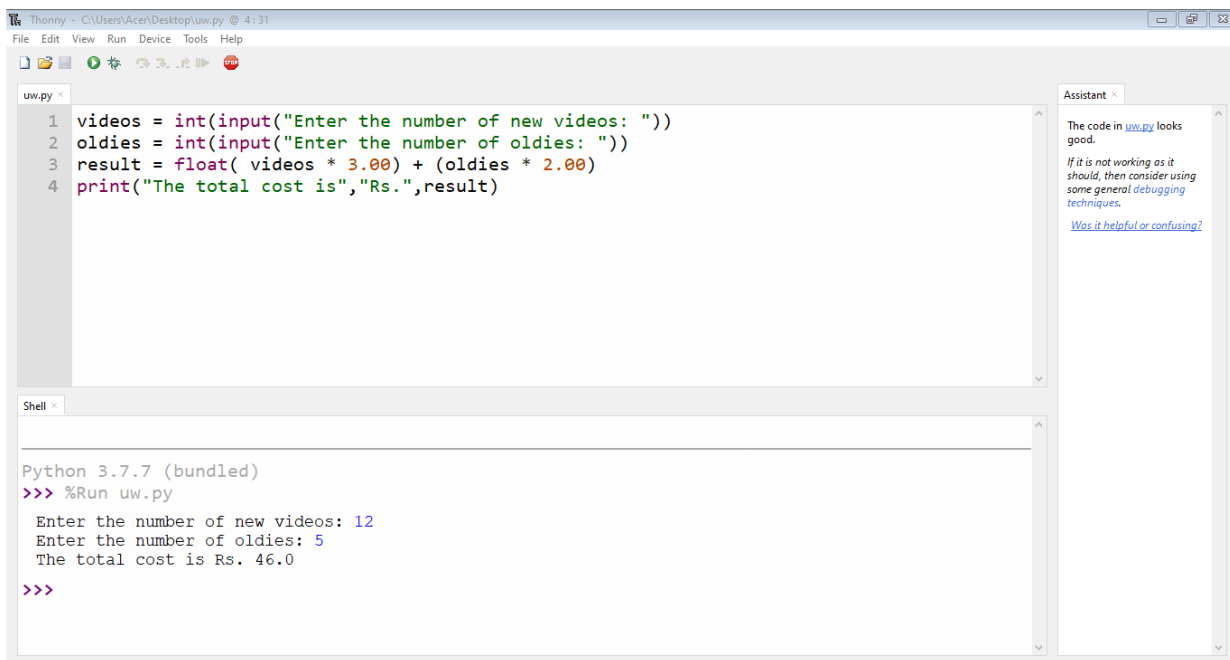
The screenshot shows the Thonny IDE interface. The top menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main editor window displays a Python script named `uw.py` with the following code:

```
1 edge = float(input("Edge Length: "))
2
3 area = 6 * edge**2
4
5 print("Surface Area: "+str(area))
```

Below the editor is a Shell window. It shows the output of running the script:

```
Python 3.7.7 (bundled)
>>> %Run uw.py
Edge Length: 12
Surface Area: 864.0
>>>
```

QN. 10



The screenshot shows the Thonny IDE interface. The top menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main editor window displays a Python script named `uw.py` with the following code:

```
1 videos = int(input("Enter the number of new videos: "))
2 oldies = int(input("Enter the number of oldies: "))
3 result = float(videos * 3.00) + (oldies * 2.00)
4 print("The total cost is", "Rs.", result)
```

Below the editor is a Shell window. It shows the output of running the script:

```
Python 3.7.7 (bundled)
>>> %Run uw.py
Enter the number of new videos: 12
Enter the number of oldies: 5
The total cost is Rs. 46.0
>>>
```

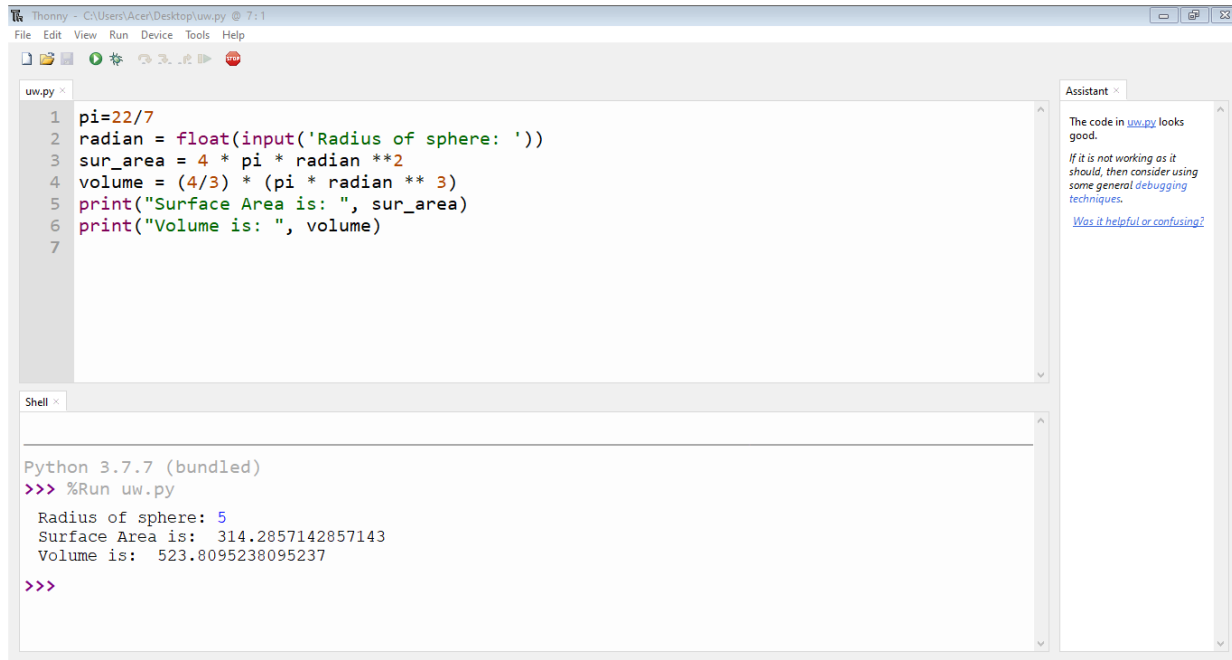
On the right side of the IDE, there is an Assistant panel. It contains the following text:

The code in [uw.py](#) looks good.

If it is not working as it should, then consider using some general [debugging techniques](#).

[Was it helpful or confusing?](#)

QN. 11



The screenshot shows the Thonny IDE interface. The main editor window displays a Python script named `uw.py` with the following code:

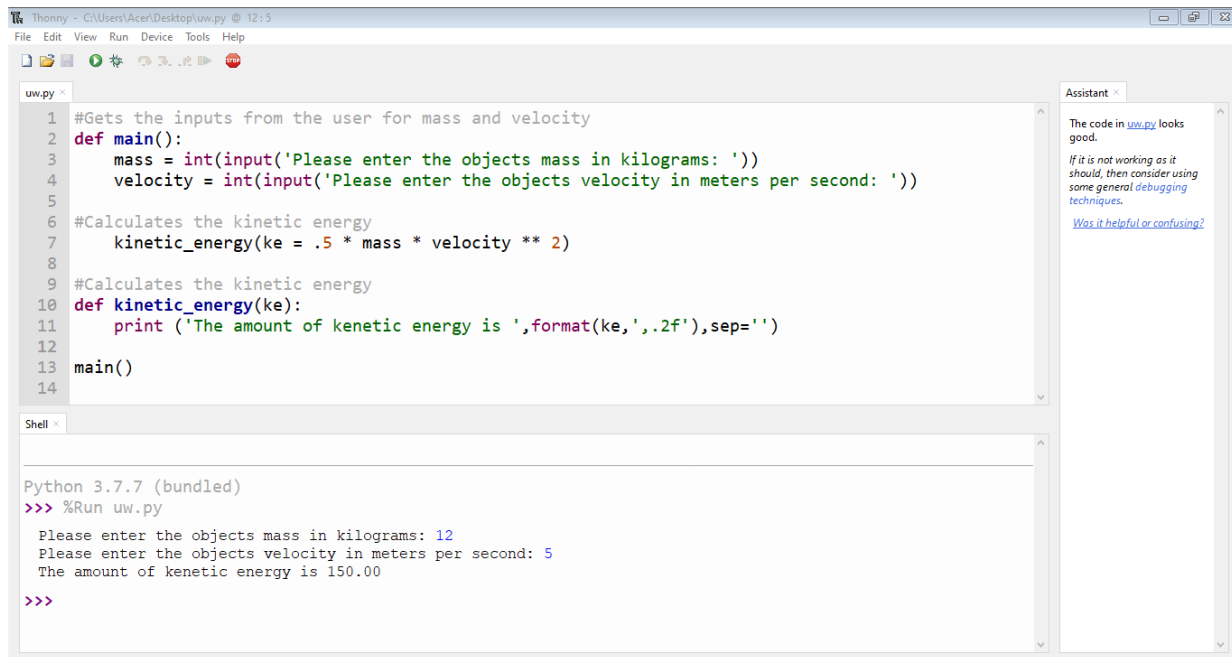
```
1 pi=22/7
2 radian = float(input('Radius of sphere: '))
3 sur_area = 4 * pi * radian **2
4 volume = (4/3) * (pi * radian ** 3)
5 print("Surface Area is: ", sur_area)
6 print("Volume is: ", volume)
7
```

The Shell window at the bottom shows the execution of the script using Python 3.7.7 (bundled):

```
>>> %Run uw.py
Radius of sphere: 5
Surface Area is: 314.2857142857143
Volume is: 523.8095238095237
>>>
```

On the right side, the Assistant panel provides feedback: "The code in `uw.py` looks good. If it is not working as it should, then consider using some general debugging techniques. [Was it helpful or confusing?](#)"

QN. 12



The screenshot shows the Thonny IDE interface. The main editor window displays a Python script named `uw.py` with the following code:

```
1 #Gets the inputs from the user for mass and velocity
2 def main():
3     mass = int(input('Please enter the objects mass in kilograms: '))
4     velocity = int(input('Please enter the objects velocity in meters per second: '))
5
6 #Calculates the kinetic energy
7     kinetic_energy(ke = .5 * mass * velocity ** 2)
8
9 #Calculates the kinetic energy
10 def kinetic_energy(ke):
11     print ('The amount of kenetic energy is ',format(ke,',.2f'),sep='')
12
13 main()
14
```

The Shell window at the bottom shows the execution of the script using Python 3.7.7 (bundled):

```
>>> %Run uw.py
Please enter the objects mass in kilograms: 12
Please enter the objects velocity in meters per second: 5
The amount of kenetic energy is 150.00
>>>
```

On the right side, the Assistant panel provides feedback: "The code in `uw.py` looks good. If it is not working as it should, then consider using some general debugging techniques. [Was it helpful or confusing?](#)"

QN. 13

The screenshot shows the Thonny IDE interface. The top bar indicates the file path is C:\Users\Acer\Desktop\uw.py. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for file operations, running, and debugging. The editor window, titled 'uw.py', contains the following Python code:

```

1 mass = float(input("Mass: "))
2 velocity = float(input("Velocity: "))
3 KE = 0.5 * mass * velocity**2
4
5 momentum = mass * velocity
6
7 print("The object's momentum is "+str(momentum))
8
9 print("The object's kinetic energy is "+str(KE))
10
11
12

```

Below the editor is a Shell window. It shows the command prompt where the script was executed:

```

>>> %Run uw.py
Mass: 5
Velocity: 5
The object's momentum is 25.0
The object's kinetic energy is 62.5
>>> |

```

Qn. 14

The screenshot shows the Thonny IDE interface. The top bar indicates the file path is C:\Users\Acer\Desktop\uw.py. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for file operations, running, and debugging. The editor window, titled 'uw.py', contains the following Python code:

```

1 hourWage= float(input ("What is your hourly wage?: "))
2
3 regularHours= float(input ("How many regular hours did you work this week?: "))
4
5 overtimeHours= float(input ("How many overtime hours did you have this week?: "))
6
7 overtimeWage= (1.5*hourWage)
8
9 totalWeeklyPay= (hourWage*regularHours)+(overtimeHours*overtimeWage)
10
11 print("Your total weekly pay is: ", totalWeeklyPay)
12

```

Below the editor is a Shell window. It shows the command prompt where the script was executed:

```

>>> %Run uw.py
What is your hourly wage?: 5
How many regular hours did you work this week?: 12
How many overtime hours did you have this week?: 5
Your total weekly pay is: 97.5
>>>

```

On the right side of the IDE, there is an Assistant panel. It contains the following text:

The code in [uw.py](#) looks good.

If it is not working as it should, then consider using some general [debugging techniques](#).

[Was it helpful or confusing?](#)